

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors: Karen P. Parnell; Hans E. E. Kedefors; Atsushi Kaneko; Daniel Salzer;
Jayant Kulkarni; Mark C. Hastings; Nikolai Tsepalov; David A. Murphy;
Gloria Lorenzo Jorge

Assignee: Siebel Systems, Inc.

Title: POLYLINGUAL SIMULTANEOUS SHIPPING OF SOFTWARE

Application No.: 09/845,785 Filing Date: April 30, 2001

Examiner: Lamont M. Spooner Group Art Unit: 2626

Docket No.: OIC0044US Confirmation No.: 4346

Austin, Texas
August 11, 2008

Mail Stop AF
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Dear Sir:

This paper is submitted in response to the Advisory Action dated July 31, 2008 and the Final Office Action dated May 12, 2008, which specified a three-month shortened statutory period for reply that ends on August 12, 2008. This Request is filed with a Notice of Appeal. No amendments are filed with this request.

Claims 1, 2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-53 are pending. Claims 1, 2, 5, 9, 16-19, 22, 26, 33-34, 38-43, and 45-53 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,442,516 issued to Lee et al. ("Lee") in view of U.S. Patent No. 6,425,123 issued to Rojas et al. ("Rojas"), and further in view of U.S. Patent No. 5,416,903 issued to Malcolm ("Malcolm"). Applicant respectfully maintains that the cited sections of the references fail to disclose each limitation of Applicant's claims.

For example, the method of independent claim 1 is usable in various implementations to facilitate shipment of a computer-implemented application. This method includes developing a base version of the application in a base language (e.g., English).

A second operation facilitates an internationalization of the base version, including a pseudo localization of language dependent code of the base version.

In various examples presented in Applicant's Specification, the pseudo localization may take the form of adding prefix characters to mark all translatable strings in the language dependent code. See, e.g., Specification, p. 18, lines 1-8. The pseudo-localization may, in various implementations, also generate a generic version of the application by removing icons, colors, or other attributes of the user interface that may be inappropriate in the locales for which the application will ultimately be translated. *Id.* at 12, lines 4-6.

A third operation facilitates a localization of the base version of the application.

In various examples presented in Applicant's Specification, the localization may use the generic version that was developed by internationalization and generate local versions by translating the marked base language (e.g., English) into other languages for the local versions (e.g., German, Spanish, French, or others). The localization in these examples may also reintroduce appropriate icons, colors, or other attributes for the various locales where the application will be used. See, e.g., Specification, p. 12, lines 6-14.

Developing the base version of the application includes developing a plurality of stages of the base version. In some examples of the method, the internationalization of a stage is performed after the development of that stage, and the localization of a stage is performed after the internationalization of that stage. See, e.g., FIG. 1 and related discussion. Thus, a step-by-step chain of events may be used in various implementations, with each stage of code passing from developers to an internationalization team that genericizes the base version, and then to translators who do localizing translations for their respective locales. These examples include a progression that is partly linear, in which internationalization builds on the developing operations, and localization builds on the internationalization.

The plain language of claim 1 makes clear that the more than a simple linear progression is required to meet the claim. The developing operation builds on the other operations: the base version of the application is modified “in response to” the internationalization or localization. The cited passages of the references fail to disclose this limitation. This modifying provides, in some implementations of the invention, iterative looping that exchanges information among the development, internationalization, and localization for the stages of an application. See, e.g., FIG. 1; p. 9, lines 4-12; and p. 25, lines 1-4.

With regard to this limitation, the Final Office Action on p. 6 cites the abstract and the following passage of Malcolm:

As a further aid in the translation process, it has been found to be extremely useful to track and log changes made during the development of the initial panels. Such changes are common in a typical engineering/software development cycle, when a product progresses through various stages prior to the end product. For example, testing may discover errors in the program, the user interface may be objected to by a human-factors specialist, etc. Yet, in order to decrease development time of products (or ‘time to market’), numerous activities must be done in parallel to reduce the overall time requirements. Therefore, a set of screen panels for a given application may need to be sent to a translation center before the final program code is completed.

Malcolm, 10:16-29 (emphasis added).

Malcolm uses a language dependent file 70 that has information with text and data that is presented in a particular language in a user interface. Malcolm, 5:54—6:9. This information is conveyed to translators or translation centers for translation into various target languages. The Final Office Action appears to equate this translation in Malcolm with Applicant’s internationalization or localization. Even if this characterization is correct (a point which Applicant does not concede), the cited passages nonetheless fail to disclose that a base version of the application is modified in response to the internationalization or localization.

The passage quoted above describes the value of tracking and logging changes in the “initial panels” that are to undergo translation in Malcolm. This cited passage teaches that various activities can be done in parallel, such as the translation of initial drafts of the panels along with further code development. This parallel division of labor is intended to decrease the

development time of products. “Therefore, a set of screen panels for a given application may need to be sent to a translation center before the final program code is completed.” *Id.* at 10:27-29. While this passage describes the forwarding of information to a translation center, it does not describe any flow of information in the opposite direction.

The Advisory Action nonetheless argues on pp. 2-3 that Malcolm teaches “a loop” because the above-quoted passage uses the word “cycle.” In particular, Malcolm makes use of the term “development cycle,” (in 10:20 and 11:31-32).

The Advisory Action argues that since Malcolm’s translators work with successive versions of translations, a previous version “has been returned to the base version developers.” Advisory Action, p. 3. Applicant respectfully disagrees.

Some of the early translation of the initial panels in Malcolm may need to be redone if code developers revise or otherwise change panels after the translators have already started converting the initial drafts into other languages. “Such changes are common in a typical engineering/software development cycle, when a product progresses through various stages prior to the end product. For example, testing may discover errors in the program, the user interface may be objected to by a human-factors specialist, etc.” *Id.* at 10:18-21.

The resulting revisions to the user interface, and other changes, cause complications for the translators who are were provided with now-obsolete versions of the initial panels. Malcolm at 10:34-56. The subsequent passages in Malcolm go on to describe how changes that affect translation can be tracked in a change log file 140. *Id.* at 10:63-68. This change log file 140 is helpful to the translation centers and translators, who may have already completed the translation of a previous version of the material. The translators can then expedite their tasks by using change log file 140 to compare the previous and current versions. *Id.* at 10:59-63; 11:26-34.

The Malcolm development cycle does not relate to any looping of operations from translation back to code development. The passage that is quoted on p. 3 of the Advisory Action, and the subsequent text, describe the passing of information (such as a language dependent file 136) “to a translation center.” See *id.* at 10:59-63. Similarly, a change log file 140 is also sent “to the translation centers.” See *id.* at 11:26-29. The cited passages do not describe any flow of information in the opposite direction, from the translation centers or translators back to program code developers.

In contrast, Applicant's claim 1 includes modifying the base version of an application. The plain language of the claim clearly states that this modifying is performed "in response to at least one of the internationalization or the localization of the base version of the application."

This limitation, in some implementations of the invention, may provide value that is not available to a practitioner of the Malcolm technology, at least because the Malcolm translators do not provide information back to Malcolm's code developers. Applicant's Specification clearly notes examples of "iterative loops" (p. 25, lines 1-4) in which, for example, "second base stage 22 is initiated, further developing the base version code, and implementing any fixes and/or changes for bugs discovered during the internationalization 14 and localization 16 of the code generated by the first base stage 20" (p. 9, lines 7-9).

At least this limitation, that the modifying is performed "in response to at least one of the internationalization or the localization of the base version of the application," is therefore absent from the cited passages of the references. At least for this reason, amended independent claim 1 and all claims dependent therefrom are allowable under § 103(a). At least for similar reasons, amended independent claims 18 and 34 and all claims dependent therefrom are also allowable under § 103(a).

Accordingly, Applicant respectfully requests withdrawal of the rejections under § 103(a).

Respectfully submitted,



Cyrus F. Bharucha
Attorney for Applicant
Reg. No. 42,324
Telephone: (512) 439-5097
Facsimile: (512) 439-5099